****

**Department of Computer Science & Engineering**

**SOUTHEAST UNIVERSITY**

**CSE4000: Research Methodology**

**FIRE AND SMOKE DETECTION WITHOUT SENSORS: IMAGE PROCESSING**

**A dissertation submitted to the Southeast University in partial fulfillment of the requirements for the degree of B. Sc. in Computer Science & Engineering .**

**Submitted by**

**Md.Kamruzzaman**

**ID : 2016100000002 Batch: 43**

**Md.Amir Hamzah**

**ID : 2016000000242 Batch: 42**

**Faridul Islam**

**ID : 2016100000077 Batch: 43**

**Supervised by**

**Samin Shahriar Tokey**

**Lecturer**

**Department of CSE Southeast University**

**Copyright Year 2020 June, 2020**

**Letter of Transmittal**

**June 22, 2020**

**The Chairman**

**Department of Computer Science & Engineering**

**Southeast University**

**Banani, Dhaka.**

**Through: Supervisor,Samin Shahriar Tokey**

**Subject: Submission of Research Report**

**Dear Sir,**

**With due respect, we have researched on Fire and Smoke Detection Without Sensors: image Processing under the course, Research Methodology. we are going to implement an Fire and smoke detection system . Actually, this is a project based research and we are trying to give here a theoretical model that we already applied.**

**So, we try our best to complete this research. we have given My best efforts to complete the research. we are requesting for your kind approval of this report. Hope you may admire our tough work and excuse the minor mistakes.**

**Thank you.**

**Sincerely yours, Supervisor:**

**Md. KamruzzamanSamin Shahriar Tokey**

**ID:2016100000002Lecturer**

**Batch:43 Department of CSE**

**Md.Amir HamzahSoutheast University**

**ID : 2016000000242 Batch: 42**

**Faridul Islam**

**ID: 2016100000077 Batch: 43**

**Certificate**

**This is to certify that the research title Implementing Fire and Smoke detection has been submitted to the respected member of the board of examiner of the faculty of Science and Engineering in partial fulfillment of the requirements for the degree of Bachelor of Science in CSE. Science and Engineering by the following student and has been accepted. This report has been carried out under our guidance.**

**Author: Approved by the Supervisor:**

**Md. Kamruzzaman Samin Shahriar Tokey**

**ID : 2016100000002Lecturer**

**Program:CSE Department of CSE**

**Faridul Islam**

**ID : 2016100000077**

**Program:CSE**

**Amir Hamzah**

**ID: 2016000000242**

**Program:CSE**

**Abstract**

**Machine learning is the science of getting computers to act without being explicitly programmed. In the past decades, machine learning has gained massive attention due to the improvement of computer hardware. Preventing fire incidents and lessening the damage of these incidents has always been a big priority. Not until very recently we were always dependent on the sensor based fire alarm to get the fire notification. As technology is evolving day by day and almost all the equipment are getting automated. After the popularity of image processing, now scientists are trying to make the computer know about different object by applying large number of positive and negative training datasets of a particular object. This same idea is the key feature of our proposed model. Our project represents a stand-alone model for simultaneous detection of fire, smoke and human from live video feed. This system applies image processing algorithm to individual segment (image). Then if any of the objects are found, it will upload the data onto cloud and send the notification to user’s phone. If multiple objects are detected the system can provide notifications for all of them in real time. Many papers have been published about different object detection's including fire and human. Our focus is to combine all the work together and to build an efficient and affordable model to tackle large scale fire outbreaks. Here people are not yet technologically literate where they can not directly interact with computers. Thinking about this, our proposed solution is very user friendly and a person with minimal knowledge will be able to operate it. We have worked on separate versions of our model to make it accessible by different classes of people..**

**Acknowledgements**

**We want to acknowledge and most grateful to Almighty God, the most merciful blessing me with patience and tenacity of mind to complete the Research paper which is the requirements for the degree successfully. Especially thanks to Shahriar Manzoor Sir,Monirul Hasan , Samin Shahriar Tokey , Department of CSE, Southeast University for selecting us among many peoples and for giving us this valuable opportunity.**

**Again we would like to thank Samin Shahriar Tokey sir who is our research supervisor giving us the opportunity to work. Without their unbelievable support, it would be quite hard to complete this Research.**

**TABLE OF CONTENTS**

**Chapter 1: Introduction ..............................................................................................................1**

**1.1 Overview...............................................................................................................................2**

**1.2 Why Image Processing? .......................................................................................................3**

**1.3 History of Various Fire Fighting Systems...............................................................................3**

**1.4 Project Aim and Objective....................................................................................................4**

**1.5 Motivation ...........................................................................................................................5**

**1.6 Thesis Outline. ................................................................................................................ …..7**

**Chapter 2: Literature Review .................................................................................................. .8**

**2.1 Overview.......................................................................................................................... …9**

**2.2 Background Studies ........................................................................................................ ….9**

**2. 3 An Existing Fire Fighting Model ...................................................................................... ….10**

**2.4 Summary ......................................................................................................................... …12**

**Chapter 3: Methodology.......................................................................................................... .13**

**3.1 Overview ......................................................................................................................... .14**

**3.2 Approach ......................................................................................................................... .14**

**3.2.1 Offline training………………………………………………………………………………14**

**3.2.2 Real time detection………………………………………………………………………...15**

**3.3 Data Collection and Processing ...................................................................................... .15**

**3.4 Training............................................................................................................................ .16**

**3.4.1 Faster R-CNN……………………………………………………………………………..16**

**3.4.2 SSD………………………………………………………………………………………...17**

**3.5 System Requirements..................................................................................................... .18**

**3.5.1 Portable version……………………………………………………………………………19**

**3.5.1.1 Raspberry PI 3 model B with cooling case………………………………………..19**

**3.5.1.2 A Standard micro SD card…………………………………………………………..20**

**3.5.2 Non-portable version……………………………………………………………………..20**

**3.5.2.1 CPU - intel core i5 7300HQ with 8GB ram………………………………………..21**

**3.5.2.2 GPU - nvidia GeForce GTX 1050 with 4GB ram………………………………...21**

**3.5.2.3 tensorflow v-1.11…………………………………………………………………….21**

**3.5.2.4 anaconda3 v-5.3……………………………………………………………………..21**

**3.5.2.5 openCV v-3…………………………………………………………………………...22**

**3.5.2.6 protobuf v-3…………………………………………………………………………...22**

**3.5.2.7 nvidia CUDA v-10…………………………………………………………………….22**

**3.6 Summary………………………………………………………………………………………......23**

**Chapter 4: Implementation...................................................................................................... .38**

**4.1 Overview.......................................................................................................................... .39**

**4.2 Cost Analysis................................................................................................................... .39**

**4.3 Installation Process ......................................................................................................... .40**

**4.4 Applications ..................................................................................................................... .40**

**4.5 Limitations ....................................................................................................................... .42**

**4.6 Summary ......................................................................................................................... .42**

**Chapter 5: Conclusion............................................................................................................. 43**

**5.1 Overview.......................................................................................................................... 44**

**5.2 Strategies and Results .................................................................................................... 44**

**5.3 Similar Works and the Scope of Improvements of Our Model........................................ 45**

**5.4 Future Work..................................................................................................................... 45**

**5.5 Summary ......................................................................................................................... 45**

**Appendix.................................................................................................................................... .46**

**Desktop Version Code:............................................................................................................47**

**Raspberry Pi Code: .................................................................................................................52**

**References................................................................................................................................ .59**

**CHAPTER 1: INTRODUCTION**

Fire, especially fire in buildings, can spread quickly andcause great loss of life and property. Therefore, early fire detection and warning is imperative. Fire detectors, smoke detectors and temperature detectors have been widely used to protect property and give warning of fires. However, smoke

and temperature detection is slower than light detection, which is the substantive detection method proposed in this paper. Furthermore, to cover the entire area potentially subject to fire,

many smoke or temperature fire detectors are required .introduces the color video to recognize the fire flame from smokes. Recent color-video based researches, propose some enhanced color image processing techniques for achieving a real-time detection offire flame. However, the above methods all focus on recognition of a fire but can't provide a reliable validation 'of a real fire and any information about whether the flame will burn up or low. This is very important when the commercial cost is considered, since human operators must manually validate each false alarm. To reduce false alarm rate in forest-fire detection, a complex hybrid system with multiple inputs provided by the visual camera, the infrared camera, meteorological sensors and a geographical information database is presented . Without losing the generality, a hybrid approach always brings a higher cost and maintenance on combination. This motivates that the attractive fire-detection method may be aimed at the general purpose, high reliable and low cost features . It uses a 2-stagedecision strategy that the first decision stage is to detect if there is a existing fire and the second decision stage is to further check whether the fire will spread out or not later on. Anyway, some fire aliases may make the first decision stage to be failed and the second decision stage needs to be refined for reducing false fire-ala" rate. To overcome the previous problems mentioned above, we improve a real fire validation by verifying the extracted fire-pixels and smoke-pixels through RGB chromatic segmentation and disorder measurement. The decisionfunction of fire-pixels is obtained by deducing with the intensity and saturation of R component, and the R, G B asompared with each other and intensity is utilized to deduce that of the smoke. Based on iterative high reliable checking of flame if a fire flame will bum up or down can be achieved. If a fire is considered to be burning up. a fire alarm will he immediately given become the fire may lead to a disaster.

* 1. **OVERVIEW**

**Fire incidents are very common all over the world. In Bangladesh, the incidents of**

**fire are more frequent. Due to the densely populated area of Bangladesh, these**

**incidents cause massive casualties and financial losses. Now-a-days, we are a**

**strong force in garments and other industrial sectors. Fire incidents in large**

**factories are one of the single most leading causes of accidental death in**

**Bangladesh. We have proposed a stand-alone system to detect fire, smoke and**

**human concurrently. The sole purpose of this model is to detect these fire**

**incidents during the intermediate breakdown and to take necessary steps**

**immediately. We have adopted an image processing approach for the detection**

**of desired objects. Our model can be integrated with any types of camera. This**

**system can not only be implemented in residential buildings but also can be used**

**in office buildings, hospitals etc to manage security. We have integrated the**

**whole system in a raspberry pi to make it portable and cost efficient. After**

**detection, the device can upload the data and give SMS notification to the desired**

**user. The user data will be uploaded in the cloud in real time so that the data can**

**be accessed from anywhere. Hence, the system can recognize fire, smoke in early**

**stage and human by processing the real time video streaming and can notify the**

**users through cloud and SMS technology. We believe that our proposed model**

**can help to mitigate the chances of large-scale fire outbreaks. Overall, this system can make our life easier and safer.**

* 1. **WHY IMAGE PROCESSING?**

**The concept of the “Internet of Things” has tied closely with the popularization of image**

**processing. IoT based system refers to the application of computer and information technology**

**for control of the devices easily from remote places. Currently, sensor based fire alarm are very**

**common. We have adopted image processing approach for few of it's advantages over sensor**

**based approach. Temperature sensor requires the fire to grow a considerable amount before it**

**can sense. Sometimes, it already causes enough damage before taking necessary steps.**

**However, in image processing approach, it can detect fire in a very early stage which helps to**

**reduce the damage. Besides, different image processing tool like OpenCV, Tensor Flow have**

**lots of built in supporting library to help us to work with large number of datasets without**

**much hassle and produce better accurate result. This IoT based system allows us to program**

**devices on the network and to remotely connect with the user through the internet. Image**

**processing provides a lot smarter and effective way of detecting fire, smoke and human.**

* 1. **HISTORY OF VARIOUS FIRE FIGHTING SYSTEMS**

**Mankind always wanted to develop system to prevent fire. The history of fire alarm is quite**

**ancient. The first automatic fire alarm was patented in 1890s by Francis Robbins Upton. This**

**timeline below explains from the early inventions of fire fighting systems leading up to today.**

**1.1902 – The invention of first heat detector: George Andrew Darby patented the first**

**electrical heat detector.**

**2. 1951 – First smoke detector was sold: Ionization smoke detectors were first sold in America in 1951.**

**They were only used in major commercial and industrial facilities for next several years due to their large size and huge cost.**

**3. 1992- The 10-year-lithium-battery-powered smoke alarm was introduced in 1995.**

**4.From 1990s to up until the present times the sensor based alarm has been the most popular**

**ones. Recently, after the popularization of machine learning concept the image processing**

**approach is being applied for the detection of different objects and its producing decent**

**accuracy.**

* 1. **PROJECT AIM AND OBJECTIVE**

**As technology is advancing, people are more safety concern now. Modern houses and**

**Industries want to secure their buildings by using various cautionary devices. In large industries**

**securing the safety of the workers should be one of the prime goals of the owner. For this**

**reason, very often they use fire alarms. The conventional fire alarms located in different parts**

**of the building and sometimes, they don't help during early fire situations. Especially, it**

**becomes more difficult when nobody's at home or the building. So, in this aspect, our proposed**

**system can help people to solve these problems as it sends the status notification to its user. By**

**using this system user can be greatly benefitted.**

**The points below show our project’s aim and objectives:**

**Our aim is to provide an easy, light and cost-efficient way of preventing large scale fire incident**

**by early detection and to notify accordingly.**

**We intend to build the system as if it will enable people who are out of their homes most of the**

**time to monitor their homes in real time from anywhere in the world by knowing whether any**

**breakdown of fire happened during their absence. Anyone can operate this system without**

**thinking much due to its simple usability and easy installation process.**

**The main objective of this project is to develop a stand-alone system which can be accessed**

**from a remote location. Using this system, users are able to know the breakdown of fire in a**

**very early situation through fire detection. Smoke detection also helps to identify large scale**

**fire nearby. These data can be sent to user's phone via Ethernet or Wi-fi.**

* 1. **MOTIVATION**

**In today’s world, large scale fire breakdown causes significant amount of casualties and**

**property loss. Bangladesh has been the victim of such incidents for many times in recent years.**

**Some of these incidents even caused more than hundred deaths and countless injuries. The**

**following table is the demonstration of recent fire incidents and its impact on the people across the country:**

**The fire incidents of Tazreen garments in 2012 took away 112 lives and hundreds of people are still carrying life threatening injuries. The most shocking fact is that in many cases after the enquiry it was founded that many of these deadly incidents could have been avoided or at least lessened the damage by taking necessary precautions. Most of the large workplaces in Bangladesh are still in crying need of necessary precautions for fire fighting. Many of the owners are often not interested to invest on these security measurements due to their high expenses and maintenance cost. Many of the household feels that they don't understand the functionality of these devices properly. All these circumstances motivated us to apply our system with existing solutions as if it doesn't complicate anything. So, we have developed two different versions of our system. One is the portable lower accuracy model and other one is the non-portable higher accuracy model. Besides with the addition of human detection we can learn whether anyone is stuck inside. Our main motivation is to save human lives and to provide a cost efficient complete system for the security of the inhabitants.**



* 1. **THESIS OUTLINE**

**The entire project is composed of six chapters, each covering a section of the work as summarized below:**

**Chapter one gives an introduction to stand alone fire fighting system as a whole, history of fire**

**fighting devices, our project’s aim and objective, it’s benefits and importance and our**

**motivation behind doing this project.**

**Chapter two covers an extensive literature review of previous works on similar systems and**

**their working procedures. Chapter three is about methodology and implementation. It provides**

**comprehensive details on both hardware components and communication services used. It also**

**explains the hardware and software requirements for this project. This section also describes**

**about applications of our model.**

**Chapter four is regarding results and testing. It gives clear practical details of testing the project**

**design and shows the output results achieved after implementing the design.**

**Chapter five is on Implementation. It includes cost analysis about our expenditure made for this**

**project. It also includes the limitations of this project that we faced during our research.**

**Chapter six is the conclusion part. There we will include the strategies that we have adopted**

**and the results we achieved. This chapter also focuses on other relevant works and the future**

**development of our model.**

**CHAPTER 2: LITERATURE REVIEW**

**2.1 OVERVIEW**

**While working on this research project, we have studied various research papers based on our project topic and selected few papers from there which were conducted on different fire fighting systems and image processing and object detection techniques. We chose these selective papers because their working approach is closely related to our research work and gathered some ideas from there for doing our work. Here, we will specifically discuss about an existing fire fighting model.**

**2.2 BACKGROUND STUDIES**

**While working on this project, we have studied various research papers based on our project top and selected few papers from there which were based on fire, smoke and human detection using artificial intelligence which uses image processing. We chose these selective papers because their working approach is closely related to our research work and gathered some ideas from there for doing our work.**

**1.The author of this article proposed a fire detection system using artificial intelligence. In this article they are suggesting about a satellite vision system where the fire can be detected from the satellite and can give immediate response to the fire fighters. The advantage of the system is that it can detect fire in open spaces and 24 hours.**

**2. In another paper, they proposed a real-time fire-detector which combines foreground information with statistical color information to detect fires. The foreground information which is obtained using adaptive background information is verified by the statistical color information which is extracted using hand labeled fire pixels to determine whether detected foreground object is a candidate for fire or not.**

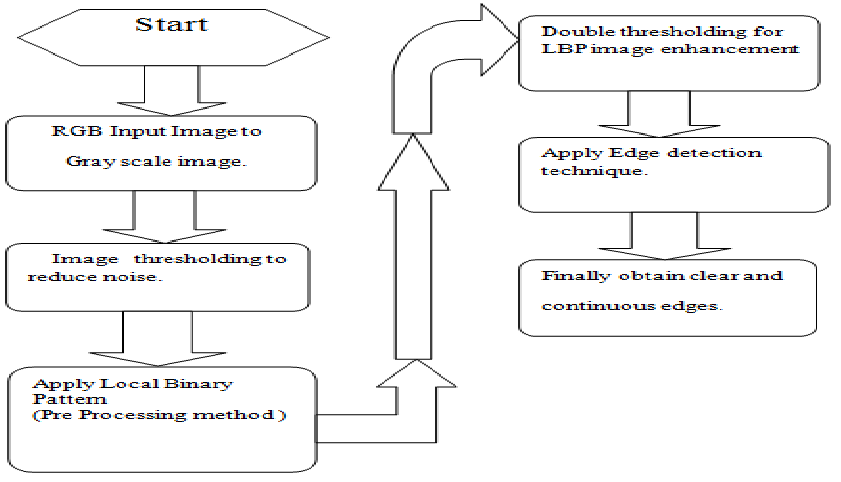
**2.3 AN EXISTING FIRE FIGHTING MODEL**

**A journal named ‘A fire fighting robot using image processing’ was published in April 2018 by Dhiraj Kumar, Dipak Kumar Diwedi and Manish Singh. In vision based mostly fireplace detection system, there are 3 major options for fire: Color pixel, moving pixels and form. The hearth pel are often classified as each in gray scale and color video sequences. It's assumed that the image capturing device produces its output in RGB color format, and these color data is employed as a pre- process step. The mixture of color and motion clues used to find the fire.**

**The block diagram of Automatic Fire Detection using GSM and image Processing consists of the following blocks:**

1. **Flame detector 2.**
2. **Temperature detector 3.**
3. **Microcontrollers**

**Flow chart:**



**2.4 SUMMARY**

**In this chapter, we have explained about other researchers’ working method on the topic of ‘fire fighting system’ and ‘object detection’ whose approach is quite similar to our project work and discussed how their working strategy helped us gain some ideas to solve some problems in our project. We have discussed about some popular image detection mechanisms that are followed current time and their short details. Here, we have also discussed about the image processing technique that we have adopted and the reason from many that we have studied.**

**CHAPTER 3: METHODOLOGY**

* 1. **OVERVIEW**

**This chapter describes the approaches of the whole system. Basically the chapter discusses about the requirements, theories, techniques and implementation of the system in brief**

**3.2.1 Offline training**

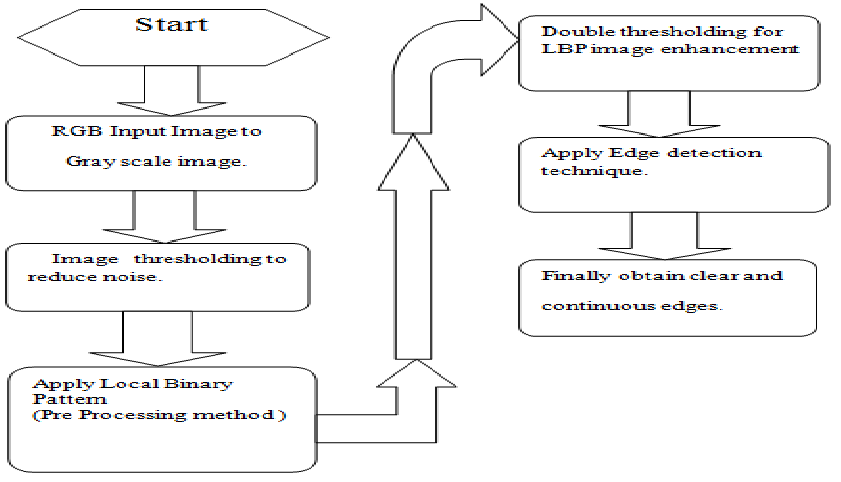
**In this section firstly, we collect the images. Then we label the images so that the object detection program can detect the labels. In this process the manual label automatically generate an xml file where the dimensions are identified. With the help of the labeled images we create our own classifier to train for detection. After the training our object detection classifier is created**.

**3.2.2 Real time detection**

**After the classifier is created, we are ready for the object detection. We take video feed from a usb camera .It also can be taken from any camera. Then we reduce the frames dimensions of the live video feed by openCV so that it matches our trained models’ dimension. Then our program start and matches the live feed with the trained model to find any object previously determined. If it finds any desired object it takes a decision and notify the user through cloud data upload and mobile SMS notification.**

* 1. **DATA COLLECTION AND PROCESSING**

**Our system need huge number of images to work and produce a standard result. For this we need standard images so that our program can work with the images. We collect our images from Adobe shutter stock and ImageNet , two open source image library. This two sources have a huge number of images . We collected around 2500 images to feed our training program**



* 1. **TRAINING**

**When we work with openCV ,we don’t train. We use some open source xml file data set to train. But that doesn’t give us our required result. So we change our approach. We collect data for training and train with some model given by tensorflow model zoo. Some of the models are,**

**faster\_rcnn\_inception\_v2\_coco**

**ssd\_mobilenet\_v1\_coco**

**ssd\_mobilenet\_v2\_coco**

**ssdlite\_mobilenet\_v2\_coco**

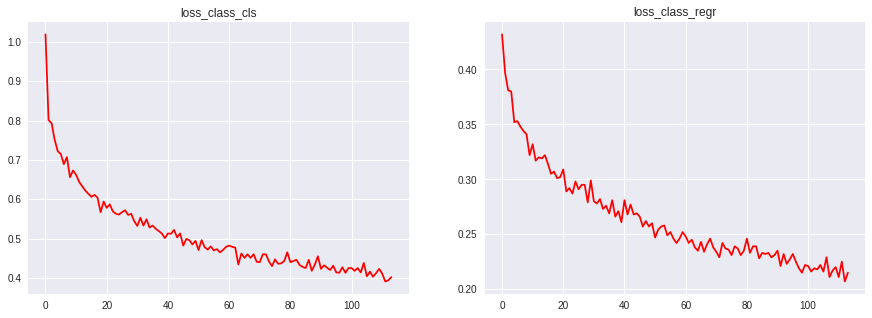
**mask\_rcnn\_inception\_v2\_coco**

**3.4.1 Faster RCNN**

**Faster RCNN replaces specific pursuit with a little convolutional arrange called Region Proposal Network to produce areas of Interests. Faster R-CNN has two**

**systems: region proposal network (RPN) for creating locale recommendations and a network utilizing these recommendations to identify objects. The 17 principle distinctive here with Fast R-CNN is that the later uses particular pursuit to create area recommendations. The time cost of producing area proposition is a lot littler in RPN than particular inquiry, when RPN shares the most calculation with the item identification organize. Quickly, RPN positions district boxes (called grapples) and proposes the ones in all probability containing objects.**

**RCNN Image:**



**3.4.2 SSD**

**The SSD architecture was published in 2016 by researchers from Google. It presents an object detection model using a single deep neural network combining regional proposals and feature extraction. A set of default boxes over different aspect ratios and scales is used and applied to the feature maps. As 18 these feature maps are computed by passing an image through an image classification network, the feature extraction for the bounding boxes can be extracted in a single step. Scores are generated for each object category in every of the default bounding boxes. In order to better fit the ground truth boxes adjustment offsets are calculated for each box**

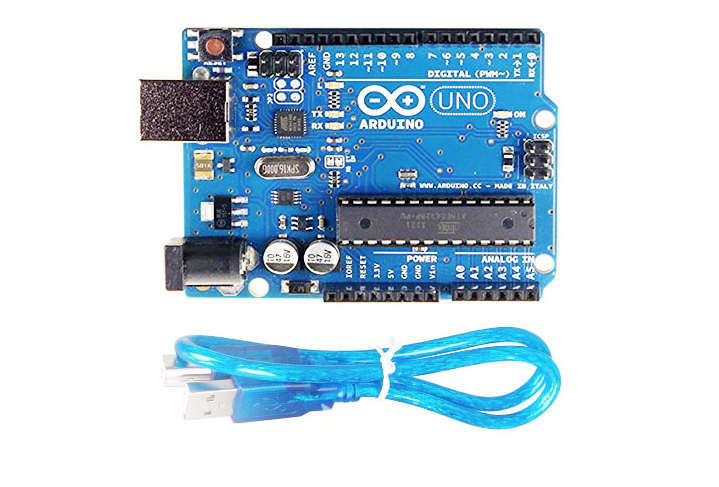
* 1. **SYSTEM REQUIREMENTS**

**We had to equip our model with necessary frameworks and drivers to implement our proposed model. We have two versions of our system; A portable version and a non-portable version. For the portable version we use**

**3.5 Raspberry PI 3**

**model B with cooling case Raspberry Pi® is an ARM-based card measured SBC(Single Board Computer) made by Raspberry Pi Foundation. Raspberry Pi runs Debian based GNU/Linux OS Raspbian and ports of various different OSes exist for this SBC**

**image arduino3.6 SUMMARY**



**This chapter shows about all the different software and hardware requirements of the whole system. It also describes about the working procedure and the systematic approach we take towards establishing the whole system. It tells us about the image collection and its processing through, labelIing, a third party software. The training mechanism along with the models used for the for both versions are describes in this chapter. In brief, the chapter shows what are the components and how they all work together to make the system as a whole**.

**CHAPTER 4: IMPLEMENTATION**

**4.1 OVERVIEW**

**In this chapter, we will clarify about - the aggregate expense of the considerable number of instruments utilized in our project, a portion of the restrictions or difficulties identified with our undertaking. This chapter also provides the installation process in details.**

**4.2 COST ANALYSIS**

|  |  |
| --- | --- |
| **Components** | **Cost (BDT)** |
| **Raspberry PI 3 model B** | **3500** |
| **Cooling case of Raspberry PI** | **400** |
| **Micro SD Card** | **600** |
| **Total** | **4500** |

**Cost of all components**

**4.3 APPLICATIONS**

**An affordable alternative of fire alarm**: At present, fire alarms are used everywhere. In big shopping malls, Universities, Factories and even in houses they are very common. They have to be set up at almost every corner of the building. It requires technicians to always look after their maintenance. For this reason, they are costly. Besides, they can't be controlled from a single platform like a monitor. In this regard, our model can play a vital role. For being very

affordable and controllable from a single source it can be worked as a direct substitute of everyday fire alarm. In addition to that, the functionality is very simple. So, once it is installed, users do not need to worry about regular maintenance. As the image processing approach helps to recognize fire in a very early stage, it gives the opportunity to turn out the fire before it gets bigger

**4.4 SUMMARY**

**Our proposed model has a simple installation process and it doesn’t require any additional skill to operate the device. We have tried to minimize the cost of the product as low as possible as if it can be used by all classes of people. We believe with further improvement, we can implement the system to be more complete and cost effective**.

**CHAPTER 5: CONCLUSION**

**5.1 OVERVIEW**

**In this chapter, we will discuss our work based on the strategies we taken and the results we achieved. We will also look into other persons did on the similar topic and what we could have done better to make a more improved system overall. This chapter also includes the proposed future work that can be done to improve the system’s efficiency**.

**5.2 STRATEGIES AND RESULTS**

**During our initial stage, we thought about a completely different system that can work on a post fire situation. We wanted to build a drone based fire fighting system to assist the fire fighter. After spending few days with the idea, we found few drawbacks of the system. First of all, it’s rather difficult to manage a good quality drone with flying capacity of more than 30 minutes in Bangladesh. The highest we found could fly only 13 minutes or so. Besides, to make a prototype we had to build a model that can be tested and won’t burn. Furthermore, the drone can’t enter all the parts of the building. As a result, we had to find a complete different strategy to work with. Then we thought about the potential value of using a system that can be integrated with the large number of cameras that are already installed. Because, it will save the users from investing on a completely new system and the cameras will be best used. This strategy, if implemented properly can fight fire in a lot earlier stage than conventional model. So far, we were able to produce a decent result with our model. Higher accuracy model can produce result that is roughly 85% accurate on average. .Portable system can detect all the three objects simultaneously with an accuracy of over 70%. Following this strategy, we can improve the accuracy of the device by working with even larger training datasets. In the preliminary testing, we could barely get the 50% detection accuracy. Then after working with larger image sets, we managed to improve the result by more than 20%.**

**5.3 SIMILAR WORKS AND THE SCOPE OF IMPROVEMENTS OF OUR MODEL**

**We have studied and found few publications that are closely related to our system in terms of strategy. Most of the other works were based on single object detection like fire or human. They were able to produce more accurate model. Many of those models provide accuracy over 98%. We lack on accuracy due to few reasons that include less powerful CPU and GPU that we trained our data with. Besides, some other objects can be added to increase functionality**.

**5.4 SUMMARY**

**Our short report represents the application of image-processing based fire, smoke and human detection system using any camera which also includes notification via cloud or SMS. Our proposed system can detect fire and smoke at early stage then notify the user so that the user can extinguish the fire easily at early stage. Our prototype will also detect if there is any human in that place where fire is detected. We believe that our system can reduce the outbreaks of fire incident. We build this system so that large fire incident like Tazreen Fashion factory in 2012 won’t happen again. In future, this product may replace fire alarms if we can improve the accuracy of the portable version of our prototype.**

**APPENDIX**

**DESKTOP AND RASPBERRY PI CODES**

**Desktop version Code:**

**# Import packages**

**from twilio.rest import Client**

**import os**

**import cv2**

**import numpy as np**

**import tensorflow as tf**

**import sys**

**# This is needed since the notebook is stored in the object\_detection folder.**

**sys.path.append("..")**

**# Import utilites**

**from utils import label\_map\_util**

**from utils import visualization\_utils as vis\_util**

**# Name of the directory containing the object detection module we're using**

**MODEL\_NAME = 'inference\_graph'**

**# Grab path to current working directory**

**CWD\_PATH = os.getcwd()**

**# Path to frozen detection graph .pb file, which contains the model that is used**

**# Path to label map file**

**PATH\_TO\_LABELS = os.path.join(CWD\_PATH,'training','labelmap.pbtxt')**

**# Number of classes the object detector can identify**

**NUM\_CLASSES = 3**

**## Load the label map.**

**# Label maps map indices to category names, so that when our convolution**

**# network predicts `5`, we know that this corresponds to `king`.**

**# Here we use internal utility functions, but anything that returns a**

**# dictionary mapping integers to appropriate string labels would be fine**

**label\_map = label\_map\_util.load\_labelmap(PATH\_TO\_LABELS)**

**categories = label\_map\_util.convert\_label\_map\_to\_categories(label\_map, max\_num\_classes=NUM\_CLASSES, use\_display\_name=True)**

**category\_index = label\_map\_util.create\_category\_index(categories)**

**# Load the Tensorflow model into memory.**

**detection\_graph = tf.Graph()**

**with detection\_graph.as\_default():**

**od\_graph\_def = tf.GraphDef()**

**with tf.gfile.GFile(PATH\_TO\_CKPT, 'rb') as fid:**

**serialized\_graph = fid.read()**

**od\_graph\_def.ParseFromString(serialized\_graph)**

**tf.import\_graph\_def(od\_graph\_def, name='')**

**sess = tf.Session(graph=detection\_graph)**

**# Define input and output tensors (i.e. data) for the object detection classifier**

**# Input tensor is the image**

**image\_tensor = detection\_graph.get\_tensor\_by\_name('image\_tensor:0')**

**# Output tensors are the detection boxes, scores, and classes**

**# Each box represents a part of the image where a particular object was detected detection\_boxes = detection\_graph.get\_tensor\_by\_name('detection\_boxes:0')**

**# Each score represents level of confidence for each of the objects.**

**# The score is shown on the result image, together with the class label. detection\_scores = detection\_graph.get\_tensor\_by\_name('detection\_scores:0') detection\_classes = detection\_graph.get\_tensor\_by\_name('detection\_classes:0')**

**# Number of objects**

**detected num\_detections = detection\_graph.get\_tensor\_by\_name('num\_detections:0')**

**# Initialize webcam feed**

**video = cv2.VideoCapture(0)**

**ret = video.set(3,1280)**

**ret = video.set(4,720)**

**account\_sid = 'AC85d35ae393a706bff28a2e182c28e6ea' # Found on Twilio Console Dashboard auth\_token = '5e70a1b2a2acc03ecdcfdb0fa3932718' # Found on Twilio Console Dashboard**

**myPhone = '+8801679118554' # Phone number you used to verify your Twilio account**

**TwilioNumber = '+16182215762' # Phone number given to you by Twilio**

**while(True):**

**# Acquire frame and expand frame dimensions to have shape: [1, None, None, 3]**

**# i.e. a single-column array, where each item in the column has the pixel RGB value**

**ret, frame = video.read()**

**frame\_expanded = np.expand\_dims(frame, axis=0)**

**# Perform the actual detection by running the model with the image as input**

**(boxes, scores, classes, num) = sess.run(**

**[detection\_boxes, detection\_scores, detection\_classes, num\_detections],**

**feed\_dict={image\_tensor: frame\_expanded})**

**# Draw the results of the detection (aka 'visulaize the results') vis\_util.visualize\_boxes\_and\_labels\_on\_image\_array(**

**frame,**

**np.squeeze(boxes),**

**np.squeeze(classes).astype(np.int32),**

**np.squeeze(scores),**

**category\_index,**

**use\_normalized\_coordinates=True,**

**line\_thickness=8,**

**min\_score\_thresh=0.60)**

**# All the results have been drawn on the frame, so it's time to display it.**

**cv2.imshow('Object detector', frame)**

**client = Client(account\_sid, auth\_token)**

**f="Fire / Human/ Smoke is detected";**

**client.messages.create( to=myPhone,**

**from\_=TwilioNumber,**

**body= f + u'\U0001f680')**

**# Press 'q' to quit**

**if cv2.waitKey(1) == ord('q'):**

**break**

**# Clean up**

**video.release()**

**cv2.destroyAllWindows()**

**Raspberry Pi Code:**

# Import packages  
import os  
import cv2  
import numpy as np  
from picamera.array import PiRGBArray  
from picamera import PiCamera  
import tensorflow as tf  
import argparse  
import sys  
# Set up camera constants  
IM\_WIDTH = 1280  
IM\_HEIGHT = 720  
#IM\_WIDTH = 640 Use smaller resolution for  
#IM\_HEIGHT = 480 slightly faster framerate  
# Select camera type (if user enters --usbcam when calling this script,  
# a USB webcam will be used)  
camera\_type = 'picamera'  
parser = argparse.ArgumentParser()  
parser.add\_argument('--usbcam', help='Use a USB webcam instead of picamera',  
action='store\_true')  
args = parser.parse\_args()  
if args.usbcam:  
camera\_type = 'usb'  
# This is needed since the working directory is the object\_detection folder.  
53  
sys.path.append('..')  
# Import utilites  
from utils import label\_map\_util  
from utils import visualization\_utils as vis\_util  
# Name of the directory containing the object detection module we're using  
MODEL\_NAME = 'ssdlite\_mobilenet\_v2\_coco\_2018\_05\_09'  
# Grab path to current working directory  
CWD\_PATH = os.getcwd()  
# Path to frozen detection graph .pb file, which contains the model that is used  
# for object detection.  
PATH\_TO\_CKPT = os.path.join(CWD\_PATH,MODEL\_NAME,'frozen\_inference\_graph.pb')  
# Path to label map file  
PATH\_TO\_LABELS = os.path.join(CWD\_PATH,'data','mscoco\_label\_map.pbtxt')  
# Number of classes the object detector can identify  
NUM\_CLASSES = 90  
## Load the label map.  
# Label maps map indices to category names, so that when the convolution  
# network predicts `5`, we know that this corresponds to `airplane`.  
# Here we use internal utility functions, but anything that returns a  
# dictionary mapping integers to appropriate string labels would be fine  
label\_map = label\_map\_util.load\_labelmap(PATH\_TO\_LABELS)  
categories = label\_map\_util.convert\_label\_map\_to\_categories(label\_map,  
max\_num\_classes=NUM\_CLASSES, use\_display\_name=True)  
54  
category\_index = label\_map\_util.create\_category\_index(categories)  
# Load the Tensorflow model into memory.  
detection\_graph = tf.Graph()  
with detection\_graph.as\_default():  
od\_graph\_def = tf.GraphDef()  
with tf.gfile.GFile(PATH\_TO\_CKPT, 'rb') as fid:  
serialized\_graph = fid.read()  
od\_graph\_def.ParseFromString(serialized\_graph)  
tf.import\_graph\_def(od\_graph\_def, name='')  
sess = tf.Session(graph=detection\_graph)  
# Define input and output tensors (i.e. data) for the object detection classifier  
# Input tensor is the image  
image\_tensor = detection\_graph.get\_tensor\_by\_name('image\_tensor:0')  
# Output tensors are the detection boxes, scores, and classes  
# Each box represents a part of the image where a particular object was detected  
detection\_boxes = detection\_graph.get\_tensor\_by\_name('detection\_boxes:0')  
# Each score represents level of confidence for each of the objects.  
# The score is shown on the result image, together with the class label.  
detection\_scores = detection\_graph.get\_tensor\_by\_name('detection\_scores:0')  
detection\_classes = detection\_graph.get\_tensor\_by\_name('detection\_classes:0')  
# Number of objects detected  
num\_detections = detection\_graph.get\_tensor\_by\_name('num\_detections:0')  
55  
# Initialize frame rate calculation  
frame\_rate\_calc = 1  
freq = cv2.getTickFrequency()  
font = cv2.FONT\_HERSHEY\_SIMPLEX  
# Initialize camera and perform object detection.  
# The camera has to be set up and used differently depending on if it's a  
# Picamera or USB webcam.  
# I know this is ugly, but I basically copy+pasted the code for the object  
# detection loop twice, and made one work for Picamera and the other work  
# for USB.  
### Picamera ###  
if camera\_type == 'picamera':  
# Initialize Picamera and grab reference to the raw capture  
camera = PiCamera()  
camera.resolution = (IM\_WIDTH,IM\_HEIGHT)  
camera.framerate = 10  
rawCapture = PiRGBArray(camera, size=(IM\_WIDTH,IM\_HEIGHT))  
rawCapture.truncate(0)  
for frame1 in camera.capture\_continuous(rawCapture, format="bgr",use\_video\_port=True):  
t1 = cv2.getTickCount()  
# Acquire frame and expand frame dimensions to have shape: [1, None, None, 3]  
# i.e. a single-column array, where each item in the column has the pixel RGB value  
frame = frame1.array  
56  
frame.setflags(write=1)  
frame\_expanded = np.expand\_dims(frame, axis=0)  
# Perform the actual detection by running the model with the image as input  
(boxes, scores, classes, num) = sess.run(  
[detection\_boxes, detection\_scores, detection\_classes, num\_detections],  
feed\_dict={image\_tensor: frame\_expanded})  
# Draw the results of the detection (aka 'visulaize the results')  
vis\_util.visualize\_boxes\_and\_labels\_on\_image\_array(  
frame,  
np.squeeze(boxes),  
np.squeeze(classes).astype(np.int32),  
np.squeeze(scores),  
category\_index,  
use\_normalized\_coordinates=True,  
line\_thickness=8,  
min\_score\_thresh=0.40)  
cv2.putText(frame,"FPS: {0:.2f}".format(frame\_rate\_calc),(30,50),font,1,(255,255,0),2,cv2.LINE\_AA)  
# All the results have been drawn on the frame, so it's time to display it.  
cv2.imshow('Object detector', frame)  
t2 = cv2.getTickCount()  
time1 = (t2-t1)/freq  
frame\_rate\_calc = 1/time1  
# Press 'q' to quit  
if cv2.waitKey(1) == ord('q'):  
57  
break  
rawCapture.truncate(0)  
camera.close()  
### USB webcam ###  
elif camera\_type == 'usb':  
# Initialize USB webcam feed  
camera = cv2.VideoCapture(0)  
ret = camera.set(3,IM\_WIDTH)  
ret = camera.set(4,IM\_HEIGHT)  
while(True):  
t1 = cv2.getTickCount()  
# Acquire frame and expand frame dimensions to have shape: [1, None, None, 3]  
# i.e. a single-column array, where each item in the column has the pixel RGB value  
ret, frame = camera.read()  
frame\_expanded = np.expand\_dims(frame, axis=0)  
# Perform the actual detection by running the model with the image as input  
(boxes, scores, classes, num) = sess.run(  
[detection\_boxes, detection\_scores, detection\_classes, num\_detections],  
feed\_dict={image\_tensor: frame\_expanded})  
# Draw the results of the detection (aka 'visulaize the results')  
vis\_util.visualize\_boxes\_and\_labels\_on\_image\_array(  
frame,  
58  
np.squeeze(boxes),  
np.squeeze(classes).astype(np.int32),  
np.squeeze(scores),  
category\_index,  
use\_normalized\_coordinates=True,  
line\_thickness=8,  
min\_score\_thresh=0.85)  
cv2.putText(frame,"FPS: {0:.2f}".format(frame\_rate\_calc),(30,50),font,1,(255,255,0),2,cv2.LINE\_AA)  
# All the results have been drawn on the frame, so it's time to display it.  
cv2.imshow('Object detector', frame)  
t2 = cv2.getTickCount()  
time1 = (t2-t1)/freq  
frame\_rate\_calc = 1/time1  
# Press 'q' to quit  
if cv2.waitKey(1) == ord('q'):  
break  
camera.release()  
cv2.destroyAllWindows()

**REFERENCES**

**[1] (n.d.). Retrieved December 25, 2018,https://patentimages.storage.googleapis.com/77/aa/c0/73df97dd063d2e/US6289331.pd**

**[2] https://www.researchgate.net/publication/247161843\_Automatic\_fire\_detection\_in\_video\_seq uences**

**[3] (n.d.). Retrieved December 25, 2018,https://www.researchgate.net/publication/329411277\_An\_Efficient\_Deep\_Learning\_Alg orithm\_for\_Fire\_and\_Smoke\_Detection\_with\_Limited\_Data**

**[4] (n.d.). Retrieved December 25, 2018,https://www.youtube.com/watch?v=Rgpfk6eYxJA&feature=share&fbclid=IwAR3e1iUf5F y1RHrACt7Bwefp5a6KbPwlrz32owX-FdtBRxpgUftRBmWkYI8**

**[5] ZHIJIAN YIN1 , BOYANG WAN1 , FEINIU YUAN2 , (Senior Member, IEEE), XUE XIA2 , AND JINTING SHI2, “A Deep Normalization and Convolutional Neural Network for Image Smoke Detection”, Digital Object Identifier 10.1109/ACCESS.2017.2747399**

**[6] Osman Gunay, Y. Hakan Habiboglu, A. Enis Centin,” Flame detection method in video using covariance descriptors”, May 2011 . Acoustics, Speech, and Signal Processing, 1988. ICASSP -88.,**

**[7] Rastislav Eštók, Miroslav Michalko, Ondrej Kainz, František Jakab,”System for Human Detection in Image Based on Intel Galileo”, (IJARAI) International Journal of Advanced Research in Artificial Intelligence, Vol. 4, No.9, 2015**

**[8] Dhiraj Kumar, Dipak Kumar Diwedi, Manish Singh,” A fire fighting robot using image processing”, April, 2018 60**

**[9] Sangeet Saha1 , Chandrajit pal2 , Rourab paul3 , Satyabrata Maity 4 , Suman Sau5, “A brief experience on journey through hardware developments for image processing and it’s applications on Cryptography”**

**[10]Turgay Çelik, Huseyin Ozkaramanli, Hasan Demirel, “Fire and smoke detection without sensors: Image processing-based approach”**

**[11]Ramnatthan Alagappan, Sourav Das,” Uncovering Twilio: Insights into Cloud Communication Services”**

**[12]Andrea C. Arpaci-Dusseau and Remzi H. Arpaci-Dusseau. Information and Control in GrayBox Systems. In Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP ’01), pages 43–56, Banff, Canada, October 2001.**